(12) UK Patent Application (19) GB (11) 2 350 993 (13) A

(43) Date of A Publication 13.12.2000

(21) Application No 9928984.5

(22) Date of Filing 09.12.1999

(30) Priority Data
(31) 10355757 (32) 15.12.1998 (33) JP

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Haruo Komooka
Kazuhiko Hasegawa

(74) Agent and/or Address for Service
R D Moss
IBM United Kingdom Limited, Intellectual Property Department, Mail Point 110, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(51) INT CL^7
G06T 15/40 15/00

(52) UK CL (Edition R)
H4T TCHD T123

(56) Documents Cited
EP 0553973 A2 EP 0455374 A2 WO 90/02990 A1
US 5923333 A

(58) Field of Search
UK CL (Edition R) H4T TBBA TBBX TCHD TCHX
INT CL^7 G06T 15/00 15/10 15/40 15/50
ONLINE DATABASES: WPI, EPODOC, JAPIO, INSPEC.

(54) Abstract Title

Generation of mixed semitransparent and opaque objects on a computer display screen.

(57) A method of semitransparent display is executable without sorting by multi-pass alpha blending for displaying a semitransparent object at a position near to the actual relative position regardless of drawing order. First, in pass 1, an opaque object alone is drawn while a hidden surface removal is executed by Z-buffer algorithm. The Z-buffer is updated. Then, in pass 2, a semitransparent object alone is blended while the hidden surface removal is executed by the Z-buffer algorithm. In this pass, the Z-buffer is not updated. Finally, in pass 3, the semitransparent object alone is blended while the hidden surface removal is executed by the Z-buffer algorithm. In this pass, the Z-buffer is updated. Although the present method needs three passes, without any need for any additional hardware, the drawing time can, however, be kept within a range of practical use, i.e. about twice the time required for the conventional method.

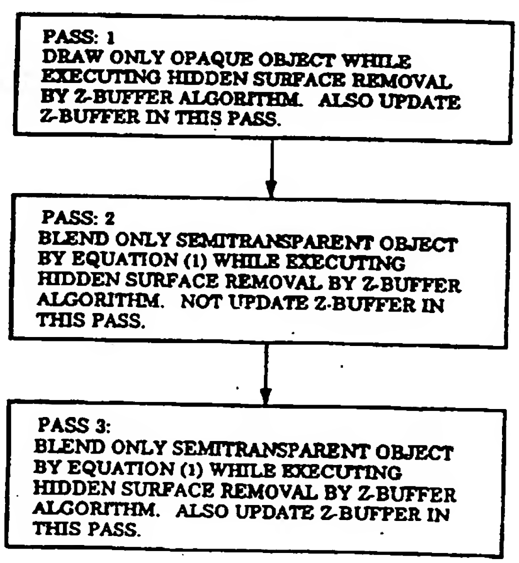
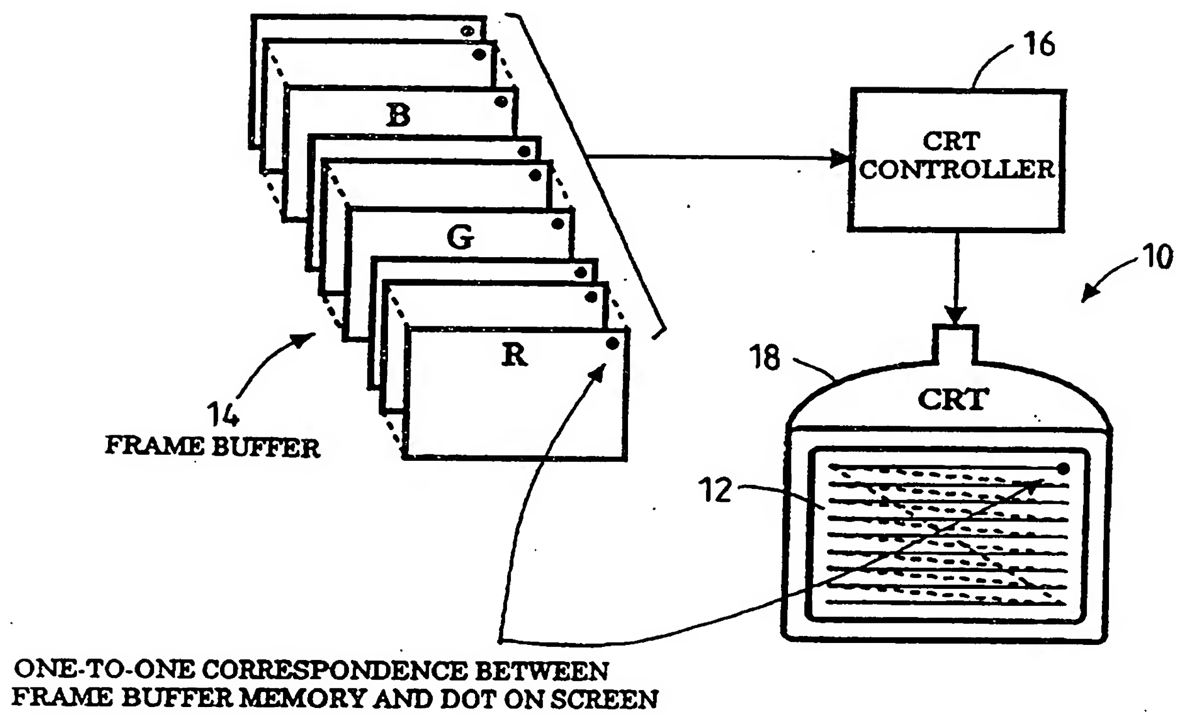


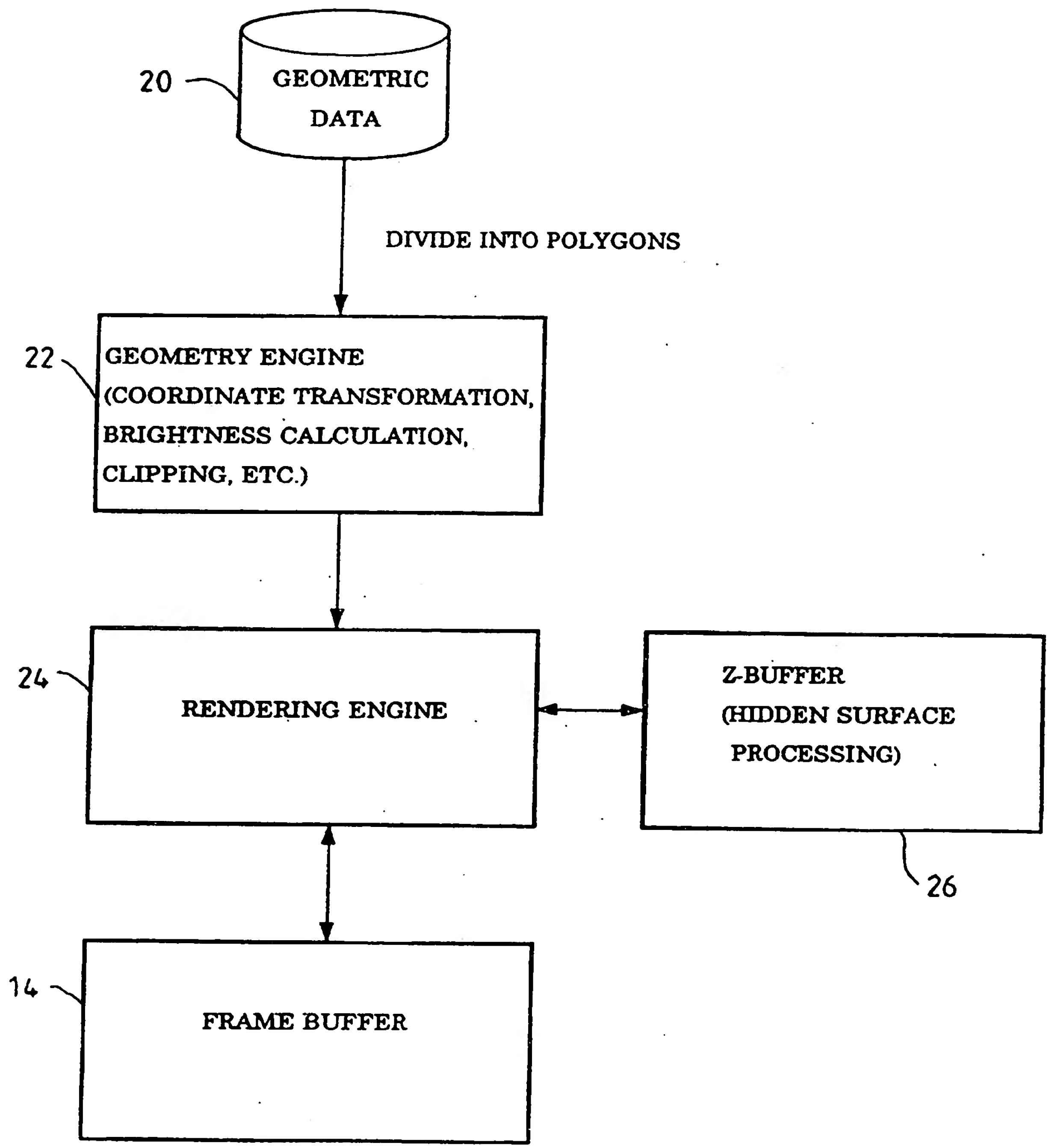
FIG. 4

GB 2 350 993 A



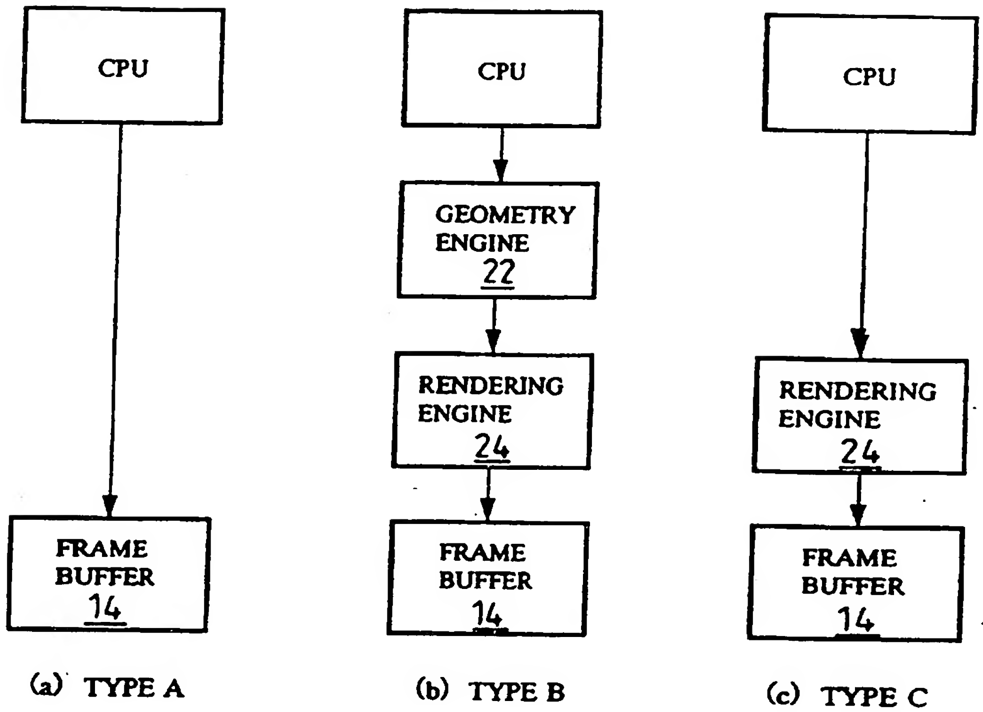
STRUCTURE OF RASTER-MONITOR CRT

FIG. 1



3D GRAPHICS PROCESSING

FIG. 2



GRAPHICS ARCHITECTURE

FIG. 3

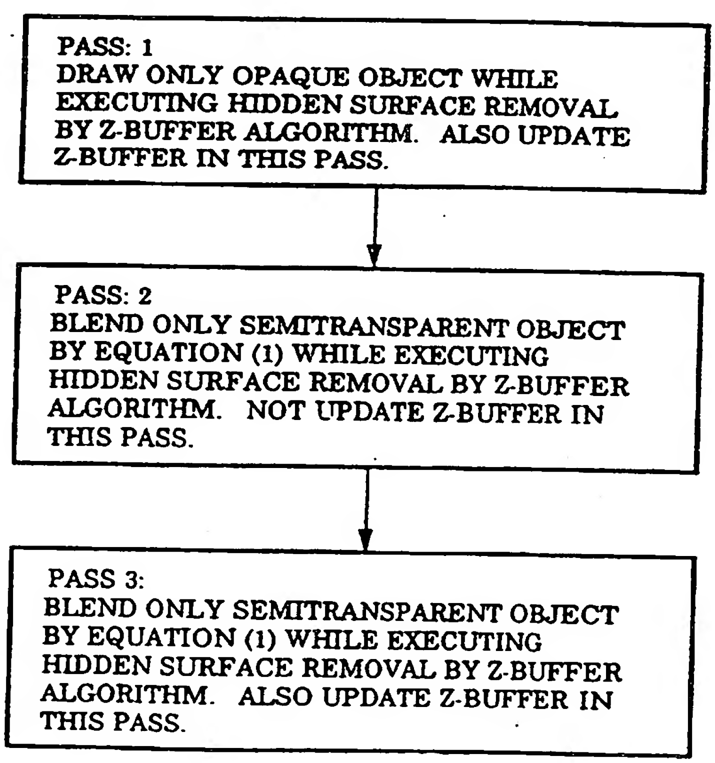


FIG. 4

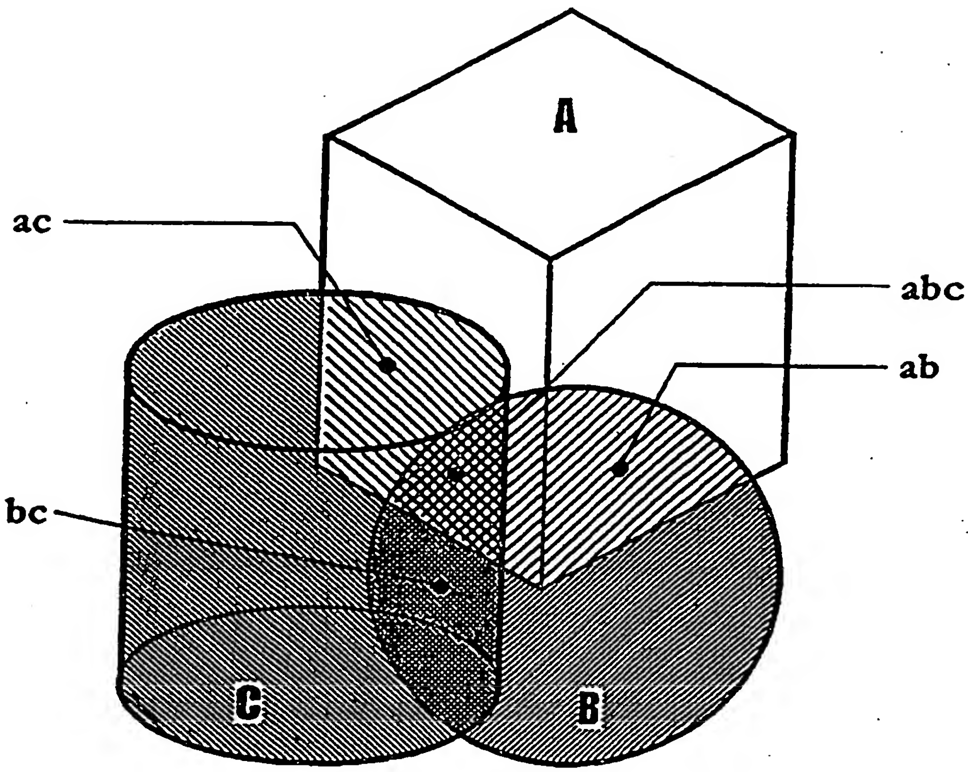
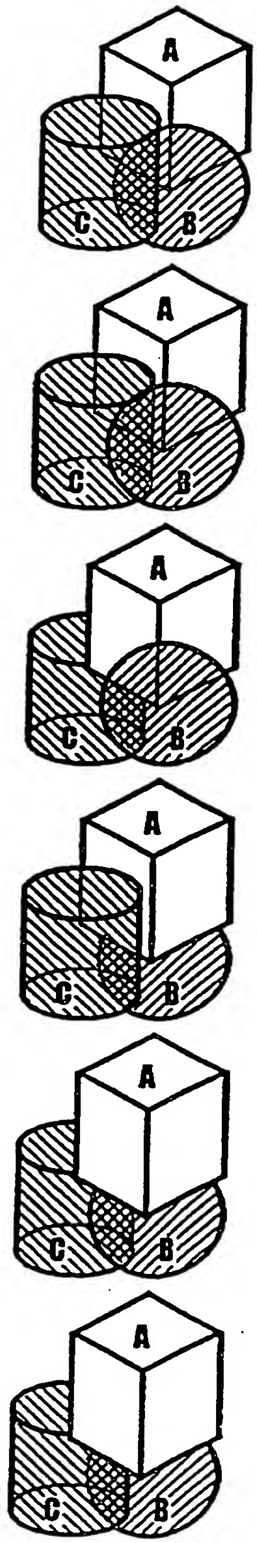


FIG. 5



C < B < A

B < C < A

B < A < C

C < A < B

A < B < C

A < C < B

FIG. 6

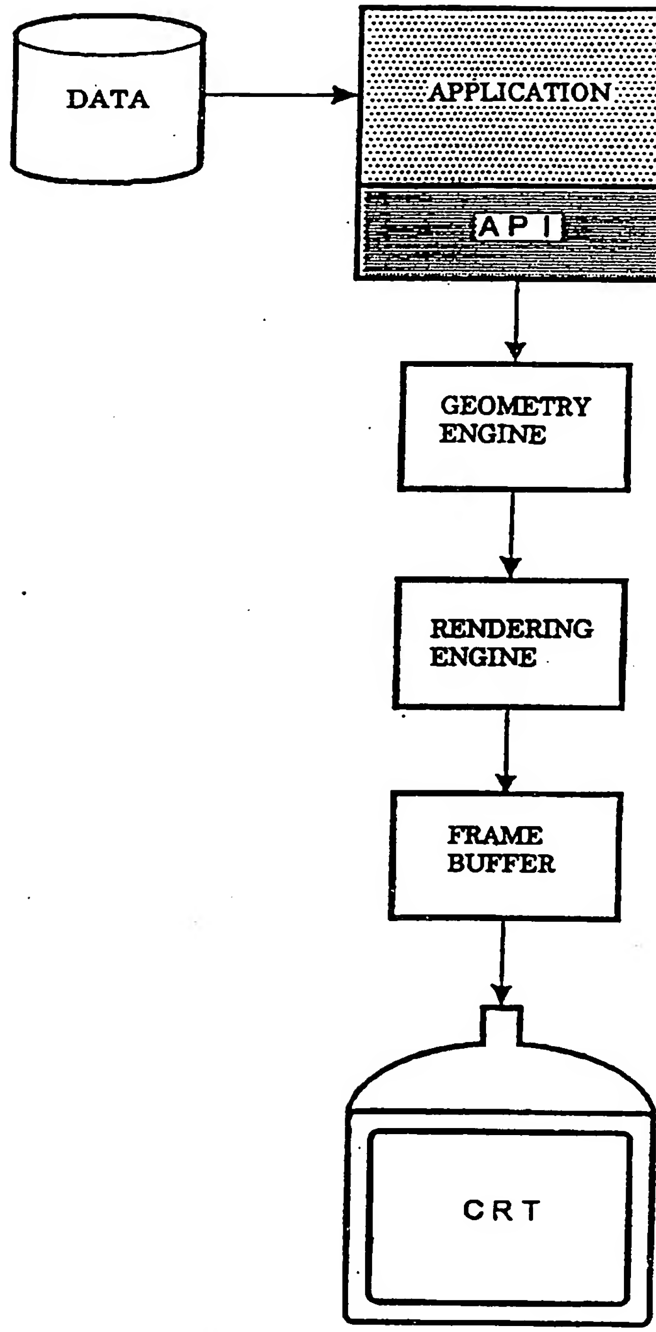

DATA
APPLICATION
API
GEOMETRY ENGINE
RENDERING ENGINE
FRAME BUFFER
CRT

FIG. 7

2350993

GENERATION OF REPRESENTATIONS OF MIXED SEMITRANSPARENT AND OPAQUE OBJECTS ON A COMPUTER DISPLAY SCREEN

Technical Field of the Invention

This invention relates to computer graphics (CG) and more particularly to the generation of representations of mixed semitransparent and opaque objects on a computer display screen.

Background of the Invention

In the field of CAD (Computer Aided Design), a three-dimensional object to be drawn must first be modeled in data. Forming a realistic object on a screen from image data representing this model is generally called rendering. Computer Graphics three-dimensional display includes modelling and rendering as main processes.

Generally, this rendering process includes shading to take into account a subtle change in shade and colour, a tinge, a reflection, a brightness, a refraction, a result of a shadow projected by a light source, a wraparound of light from the periphery, a picture, a transparency, an opacity, or the like, in order to reproduce the appearance and colour of the surface of the object more realistically just like a photograph.

In the rendering process, the model of each three-dimensional object is finally projected from a three-dimensional object space onto a screen that is a plane of projection for a two-dimensional image space. at this time, the model is projected and displayed in two dimensions, i.e., onto a computer display screen (the plane of projection) by a perspective projection (central projection) from a view point or a centre of projection or by a parallel projection from infinity in accordance with information on depth normal to the screen (also called "Z-direction") in such a manner that the screen is drawn or painted over to represent the object in accordance with the model.

The model is projected and displayed on the screen so that the screen is drawn according to the model, by either (1): directly writing (W) the image data of the object displayed in a storage area prepared as a display buffer in storage, or (2): first reading (R) the image data stored in the buffer storage area; then modifying (M) the read image data; and then writing (W) the modified image data (RMW cycle). This is a general cycle for "drawing" an object.

For such a projection, a hidden surface removal algorithm called Z-buffer algorithm is well known as an approach which can most simply execute hidden line removal and hidden surface removal (hereinafter collectively referred to as hidden surface removal). Imagine pixels, each individual pixel being capable of having the colour and brightness allocated thereto, each pixel being a minimum unit of a display plane. In the Z-buffer algorithm, as its name signifies, each pixel has a memory location (depth memory) for storing its Z-value (a distance from the view point, the centre of projection or the like, or the depth information) of the object. As a whole, there is need for a memory array (or "Z-buffer") for the number of pixels employed.

The display progresses by painting over the screen in an opaque colour alone while the Z-buffer algorithm is used to achieve complete hidden line removal and hidden surface removal. However, it is then impossible to see a deep portion of the object which is obscured by an opaque colour. In short, the Z-buffer algorithm is used in the display of opaque objects. On the other hand, the field of CAD also requires a displaying method in which not only the surface but also an interior portion can be simultaneously viewed by semitransparently displaying the object.

One approach for semitransparent display is an approach using blending. In this approach, coefficients A and B are used for the colour (src_colour) of the object to be drawn and the colour (dest_colour) of the object that is already drawn, respectively. A new colour (colour) is determined by colour=A•src_colour+B•dest_colour. More particularly, the colour (dest_colour) of the object that is already drawn is read (R). Then, the colour is modified (M) by the colour (src_colour) of a certain object. Then, the new colour (colour) is written (W). In other words, the read colour (dest_colour) of the object is replaced by the new colour (colour), so that the new colour (colour) is displayed. In this case, the modification of the new colour (colour) is calculated by multiplying the coefficients A and the coefficient B.

Particularly, equation (1) is generally used as a method (alpha blend) using a semitransparent coefficient alpha (sometimes called alpha blending rate).

$$\text{colour} = \text{alpha}\cdot\text{src\_colour} + (1-\text{alpha})\cdot\text{dest\_colour} \quad (1)$$

Calculating and drawing the new colour by using equation (1) is referred to as "blending" below. Blending has the following characteristic. At the same point, the object that is later drawn seems nearer to the view point than the object that is already drawn.

Consequently, an attempt at the exact semitransparent display by this approach needs to use a priority list method. That is, the objects are sorted in accordance with the distance from the view point of the user (i.e., in accordance with the depth along the depth direction), whereby the priority of the object to be drawn is determined and the priority list is created. The objects must be drawn in the decreasing order of the distance from the view point in accordance with the list priority.

If the objects are drawn by using this approach without determining the priority by sorting, the relative near and far positions of the objects may appear to be displayed differently from their actual positions. This is the phenomenon that the object that is later drawn is seen nearer regardless of the relative positions of the objects. This is a problem which arises when the object near to the view point is drawn early and the object far from the view point is blended later. Accordingly, since the objects are not sorted, the objects are not always drawn in the desired order and thus the display of a desired appropriate enhancement is not ensured.

Thus, seeking a high quality for ensuring a strict perspective drawing for the user needs strict sorting for the semitransparent display. However, the larger the number of objects to be sorted is, the more time is required for the sorting. In this situation, the processing speed required for the whole graphic is considerably reduced. This is a serious problem. As can be seen from a rough estimate, a number of processing steps, nlog(n) is generally needed in order to sort a permutation $_nP_n$ (n denotes the number of objects to be sorted).

The use of this sorting procedure is disadvantageous to an application such as CAD which requires interactive real-time data editing. More particularly, the sorting of every pixel is required to edit a plurality of free curved surfaces, when one pixel represents one of the facets which the curved surface is divided into. Assuming the general screen having 1280 x 1024 pixels, the decreased processing speed would be no longer acceptable in practice.

Disclosure of the invention

If there were no significant visual problem for the user, alpha blending could be simply and quickly performed even without the complete sorting. Furthermore, the ability to omit the sorting and thus realize high-speed processing would be very preferable, because the processing speed is important for the application for the interactive data editing.

Accordingly, the present invention provides a method of multi-pass semitransparent processing having three passes, being a drawing method for displaying image data about a plurality of objects including opaque objects and semitransparent objects, each having information about a depth direction, on a computer display screen by using an updatable Z-buffer as a storage, the method comprising the steps of: drawing the opaque object alone of the image data, while updating the Z-buffer and executing a hidden surface removal by the Z-buffer algorithm; drawing the semitransparent object alone of the image data without updating the Z-buffer and while executing the hidden surface removal by the Z-buffer algorithm; and drawing the semitransparent object alone of the image data, while updating the Z-buffer and executing the hidden surface removal by the Z-buffer algorithm.

As a hardware resource capable of implementing the multi-pass method of the present invention, there is provided a drawing apparatus which can display image data about a plurality of objects including opaque objects and semitransparent objects, each having information about a depth direction, on a computer display screen, the apparatus comprising: an updatable Z-buffer corresponding to each dot on the computer display screen, for storing the depth information; an updatable frame buffer corresponding to each dot on the computer display screen, the frame buffer being capable of outputting the contents therein to the display screen, for storing the data to be displayed; and a rendering engine, the engine receiving the image data and then comparing the depth information on the image data to be now drawn to the depth information that is already stored in the Z-buffer, thereby permitting judging whether or not the depth information on the image data to be now drawn is larger than the depth information that is already stored in the Z-buffer, the engine being capable of selecting either outputting the data to be displayed while updating the depth information or outputting the data to be displayed without updating the depth information, and the engine being capable of reading the data to be displayed that is already stored in the frame buffer and then blending the read data with the image data received thereafter for each dot on the computer display screen.

Brief Description of the Drawings

The invention will now be further described, by way of example only, with reference to preferred embodiments thereof as illustrated in the accompanying drawings, in which:-

Fig. 1 shows a raster monitor CRT system;

Fig. 2 is a flow chart showing the steps of general graphic processing for obtaining graphic information to be transferred to a frame buffer such as that of Figure 1;

Fig. 3 shows three alternative computer architectures for implementing graphics processing;

Fig. 4 is a flow chart showing multi-pass processing of graphics data according to the invention;

Fig. 5 illustrates in large scale the display result when three objects A, B and C are drawn in order A, B and C in three passes, according to the method of the present invention;

Fig. 6 illustrates on a smaller scale the display result when the three objects A, B and C are drawn, for all possible relative positions of the three objects according to the method of the invention; and

Fig. 7 illustrates how a method according to the present invention can be implemented at the application level.

<u>Detailed Description of the Invention</u>

Fig. 1 shows a hardware configuration for processing computer graphics. A raster monitor CRT 10 having a cathode-ray tube (CRT) for refreshing data is described. However, it is very easy for one skilled in the art to substitute any other display such as a liquid crystal display (LCD) instead of the CRT.

Geometric information to be displayed on a screen 12 is digitized. The digitized data is temporarily stored, dot by dot, in a frame buffer memory 14. The dot-by-dot information in the frame buffer is composed of red, green and blue brightness information (if 8(n) bits were given, number of status $2^8(2^{(n)})$ can be represented.) or information equivalent to the brightness information. One pixel corresponds to one dot in order to have a one-to-one correspondence between the pixel and the frame buffer memory 14.

The digitized data in the frame buffer memory is transferred to a CRT controller 16. The data is converted into a video signal. An electron beam is oscillated and moved on a scanning line in CRT 18, whereby the desired graphic information is displayed on the screen 12.

Fig. 2 is a flow chart showing general graphics processing for obtaining the graphic information to be transferred to the frame buffer. Geometric data 20 modelling an object or an object to be drawn is divided

into polygons and then transferred to a geometry engine 22. The geometry engine 22 calculates the brightness of a vertex from a coordinate transformation of the vertex of the polygon, a clipping, a light source, the information about a material of the object, or the like. A rendering engine 24 calculates a colour of each pixel by interpolation in accordance with the vertex data and the brightness information. The data undergoes a hidden surface processing by Z-buffer 26. The data is then written to the frame buffer.

The Z-buffer, hidden surface processing, hidden line processing, shading or the like can be also executed by a high-speed graphics accelerator having a graphics processor for use in graphics only, independent of CPU.

Fig. 3 shows three types of computer architecture for implementing the graphics processing. Type A causes CPU of a workstation to execute all the processing in software and has the frame buffer 14 alone as hardware for the display. Type B has a pipeline structure including the geometry engine 22 and the rendering engine 24, all in the form of hardware. Type C has the rendering engine 24 in hardware but a program in the CPU executes the processing to be executed by the geometry engine 22.

The type B or C is used for a complicated processing handling a great deal of data. A drawing method according to the present invention can be implemented by using a display processor of any one of types A, B and C, as long as the architecture is provided with a Z-buffer (or any pseudo component that can be controlled by the software) capable of independently controlling a comparison and an updating for a hidden surface removal, and a frame buffer having alpha blending function. More particularly, this method is useful for the types B and C having the hidden surface removal and the alpha blending function carried out in hardware.

The Z-buffer algorithm does not follow list priority and its advantage is characterized in that new objects can be additionally drawn one after another even after the drawing of all the objects is once finished. The Z-buffer algorithm does not hold the information about the objects that are drawn before or behind the object that is now being drawn, because the object is drawn one at a time (regardless of the other objects). A very small area is therefore enough for the Z-buffer memory storage area.

However, the Z-buffer algorithm is also characterized in that an object once drawn cannot be removed. The reason is as follows. In the Z-buffer algorithm, whether or not the other objects are in a model space

has no effect on the procedure of the hidden surface removal. In other words, the object which must be hidden and thus unseen may be drawn at any point in the processing. This is caused by the following fact. That is, the processing only requires data about the object to be newly drawn and the contents of the Z-buffer alone.

The Z-buffer algorithm is processed in the following manner. The updated value is used for the subsequent drawing, while the Z-buffer is sequentially updated. If the Z-buffer is not updated (i.e., if the updating is stopped), the intact value can be maintained and used.

In the present invention, the characteristic of the Z-buffer algorithm that the new object can be additionally drawn later is therefore used in successful combination with the approach for the semitransparent display using the alpha blending, in accordance with the timing of the updating of the Z-buffer. An advantage is thereby obtained in the omission of the priority list sorting procedure which is accurate but takes time, by using the combination.

That is, the idea that the objects are first sorted in accordance with the distance so that a semitransparent object is exactly displayed, is abandoned. However, attention is directed to the following conditions 1 to 3. These conditions are important for displaying the semitransparent object at the position near to the actual relative position regardless of the drawing order. These conditions are carefully set not to cause a visual problem for a user.

Condition 1: The object nearest to the view point should be last blended or drawn regardless of whether the object is semitransparent or opaque. This is attributed to the following characteristic. The object that is last alpha-blended or drawn is seen nearer to the view point than the object that is previously alpha-blended or drawn, regardless of whether the object is semitransparent or opaque.

Condition 2: An object that is farther from the view point than an opaque object is not displayed, regardless of whether it is semitransparent or opaque. This is based on the characteristic that the opaque object obscures the object located behind the opaque object.

Condition 3: An object that is farther from the view point than a semitransparent object is alpha-blended and displayed regardless of whether it is semitransparent or opaque, except for the case where the object is not displayed under the condition 2.

These conditions 1 to 3 are satisfied by multi-pass processing (also called "multi-pass semitransparent processing" because it is

characterized by the processing of the semitransparent object) which is divided into three steps as described below, instead of the sorting procedure that has been needed.

Pass 1: Only the opaque object is drawn while the hidden surface removal is executed by the Z-buffer algorithm. In this case, the Z-buffer is also updated.

Pass 2: Only the semitransparent object is blended by equation (1) while the hidden surface removal is executed by the Z-buffer algorithm. In this case, the Z-buffer is not updated.

Pass 3: Only the semitransparent object is blended by equation (1) while the hidden surface removal is executed by the Z-buffer algorithm. In this case, the Z-buffer is also updated.

In the "one-pass" method of the prior art, the data has been drawn in the given order without the hidden surface removal by the Z-buffer algorithm and without the data sorting. In multi-pass, the data is drawn through three passes.

A function of each of the passes 1 to 3 will be described in further detail.

In the pass 1, only the opaque object is processed. At this time, the hidden surface removal is executed by the Z-buffer algorithm and the Z-buffer is updated in the same manner as the normal processing of the opaque object. The alpha blending by equation (1) is not executed. As a result, the opaque object nearest to the view point is displayed. The value of the Z-buffer is changed to the corresponding value. As for the opaque object, the condition 2 is satisfied, because the opaque object is not drawn in the passes 2 and 3.

In the pass 2, only the semitransparent object is processed. The alpha blending by equation (1) is executed while the hidden surface removal is executed by the Z-buffer algorithm. The Z-buffer is not updated. Only the object, which is nearer to the view point than the opaque object drawn in the pass 1, is consequently alpha-blended because of the hidden surface removal by the Z-buffer algorithm. Also, for the semitransparent object, the condition 2 is satisfied, because the Z-buffer was updated to the depth value of the opaque object in the pass 1 and thus the semitransparent object farther from the view point than the opaque object is not drawn. The semitransparent object, which is nearer to the view point than the opaque object drawn in the pass 1, is always blended, because the Z-buffer is not updated in this pass 2. As a consequence, the condition 3 is satisfied.

At the end of the pass 2, the value of each point in the Z-buffer is kept at the same value as the value which is obtained at the end of the pass 1. That is, also at this time, the Z-buffer holds the distance to the opaque object nearest to the view point on each point.

In the pass 3, only the semitransparent object is again processed. The alpha blending by equation (1) is executed while the hidden surface removal is executed by the Z-buffer algorithm. The Z-buffer is also updated. The difference between the passes 2 and 3 is therefore that the Z-buffer is updated. In the pass 2, both the two semitransparent objects are alpha-blended regardless of the relative positions of these objects, because the Z-buffer is not updated.

During the processing in the pass 3, the Z-buffer holds the distance from the view point to the object that is last alpha-blended or drawn on each point. The object to be last alpha-blended is therefore the object nearest to the view point. The condition 1 is thus satisfied. When the semitransparent object does not exist between the opaque object and the view point, the object drawn by the pass 1 is last drawn. Also in this case, the condition 1 is satisfied.

It should be noted that although the relative positions of the (opaque or semitransparent) object nearest to the view point and the other (opaque or semitransparent) objects are seen without any trouble, it is not always certain that the relative position of the semitransparent object situated between these objects is seen. This is attributed to the omission of the exact sorting procedure.

The relative positions of objects other than the nearest object depend on the order of the drawing by an application. For example, a typical interactive processing, CAD often includes rotating the object so as to see it from the opposite direction. It is impossible to define data in order of the distance from the eye position of the user unless the objects are sorted. The positions of these objects are, however, less important than the position of the (opaque or semitransparent) object nearest to the view point, and thus they can admit of a large compromise.

How "drawing" can be executed will be described in connection with RMW cycle. The drawing in the pass 1 does not need to execute all of R, M and W. The execution of W alone is sufficient for the replacement by the image data about the opaque object. However, the drawing in the passes 2 and 3 needs to blend the current background colour (the colour (dest_colour) of the object that is already drawn) with the colour of the semitransparent object to be now drawn (the colour (src_colour) of the object to be drawn). Thus, it is necessary to execute all of R, M and W.

As a more particular example, three objects to be drawn, i.e., an opaque object A, a semitransparent object B and a semitransparent object C are assumed. The display result will be discussed which is given by the passes 1 to 3, the approach of the present invention, when the objects are always drawn in the order of A, B and C by an application program for graphics drawing.

Herein, A<B represents that A is nearer to the view point than B by the use of a sign <. A portion, in which three objects A, B and C overlap when seen from the view point, will be discussed below.

C<B<A (C is nearest to the view point and A is farthest from the view point):

In the pass 1, A is drawn.
In the pass 2, B is blended and then C is blended.
In the pass 3, B is not blended and then C is blended.
As a result, C is seen nearest and B and A are seen behind C.

Fig. 5 illustrates the drawing result which is obtained in the case of C<B<A (C is nearest to the view point and A is farthest from the view point), in more detail (than the drawing result of Fig. 6 described below). In this drawing, the portion associated with the object B is represented by slanted lines (extending from the upper right to the lower left), all of which are directed in the same direction. The portion associated with the object C is represented by the slanted lines (extending from the upper left to the lower right), all of which are directed in the same direction that is different from the direction of the lines representing the object B. A region to be drawn as mentioned below is specified by not only one type of slanted lines but also a combination of two types of slanted lines crossing. More particularly, a mesh-like region to be drawn also exists in which the slanted lines extending from the upper right to the lower left and the slanted lines extending from the upper left to the lower right cross.

A region ab to be drawn is the portion in which A and B overlap. This region is drawn in such a manner that A is seen behind B. A region bc to be drawn is the portion in which B and C overlap. This region is drawn in such a manner that B is seen behind C. A region ac to be drawn is the portion in which A and C overlap. This region is drawn in such a manner that A is seen behind C. Furthermore, a region abc to be drawn is the portion in which all of A, B and C overlap. This region is drawn in such a manner that B is seen before A and C is seen before B. The difference between the region bc to be drawn and the region abc to be drawn is represented by a mesh density which varies depending on whether or not the object A and the objects B and C overlap.

Fig. 6 shows all the possible relative positions of the three objects A, B and C including C<B<A. As for cases other than C<B<A, the detailed representation and description of the drawing result shown in Fig. 5 are omitted.

B<C<A (B is nearest to the view point and A is farthest from the view point):

In the pass 1, A is drawn.

In the pass 2, B is blended and then C is blended.

In the pass 3, B is blended but C is not blended.

As a result, B is seen nearest and C and A are seen behind B.

B<A<C (B is nearest to the view point and C is farthest from the view point):

In the pass 1, A is drawn.

In the pass 2, B alone is blended, and C is hidden behind A and thus C is not blended.

In the pass 3, B alone is blended, and C is hidden behind A and thus C is not blended.

As a result, A is seen behind B.

C<A<B (C is nearest to the view point and B is farthest from the view point):

In the pass 1, A is drawn.

In the pass 2, only C is blended, and B is not blended because B is hidden by A.

In the pass 3, only C is blended, and B is not blended because B is hidden by A.

As a result, A is seen behind C.

A<B<C (A is nearest to the view point and C is farthest from the view point):

In the pass 1, A is drawn.

In the pass 2, nothing is drawn.

In the pass 3, nothing is drawn.

As a result, A alone is displayed.

A<C<B (A is nearest to the view point and B is farthest from the view point):

In the pass 1, A is drawn.

In the pass 2, nothing is drawn.

In the pass 3, nothing is drawn.

As a result, A alone is displayed.

In the above examples, A is first drawn. However, the opaque object is always processed earlier than the semitransparent object

regardless of the order of the drawing by the application program, because the opaque object is processed in the pass 1 and the semitransparent object is processed in the passes 2 and 3. The same result is therefore obtained in the case where the opaque object is second or third drawn.

As described above, the approach of the present invention is not used for exact semitransparent display. However, the following fact was confirmed. When the alpha blending is used in the multi-pass method according to the present invention, the objects are displayed in such a manner that the near and far relative position is kept, compared to the case where the alpha blending is used in only the one-pass.

In Table 1, there is shown the result of measurement of performance in an actual application. The measurement result of the semitransparent drawing with the sorting is not obtained, because it has no practical use and it is difficult to implement. Therefore, the result of a simulation is measured as the time required for a logic processing for the sorting. This result is added to the semitransparent processing time in the one-pass, whereby the estimated time is determined and used as the time to be compared. This is judged to be unusable for the interactive processing.

Table 1

Time required for drawing repeated 100 times by different drawing approaches

	Drawing approach	Benchmark 1	Benchmark 2
[1]	Opaque	8.3 sec	64.6 sec
[2]	One-pass semitransparent	8.3 sec	63.9 sec
[3]	Estimated sorting time	272.8 sec	767.0 sec
[4]	Semitransparent after sorting	281.1 sec	830.9 sec
[5]	Multi-pass semitransparent	15.0 sec	129.2 sec

In more particular, [1] and [2] are drawn by using a current IBM product known as graPHIGS (graPHIGS is a trademark of IBM Corp. in U.S. and other countries) which is a three-dimensional (3D) graphics API (Application Programming Interface). IBM Corp. implements the ISO Standard "PHIGS" (Programmer's Hierarchical Interactive Graphics System) as graPHIGS. [2] is the semitransparent processing by the one-pass blending without the sorting. In this case, the relative position is not retained. In [3], the time required for the sorting is measured and estimated by a simulation program. [4] is determined by assuming that [4]=[2]+[3]. [5] is that the current graPHIGS is remodelled and implemented so that the multi-pass semitransparent processing of the present invention can be executed whereby the object is drawn.

[1] and [2] can be processed in a single pass. On the other hand, [5] to which the present invention is applied needs three passes 1 to 3. The drawing time is, however, within a range of more practical value, i.e., it is about twice the time required for the opaque drawing and conventional method. User evaluation of drawing as in step 5, using actual data resulted in full satisfaction with the visual relative position of displayed objects and the drawing speed for interactive processing.

As can be seen from a relationship as shown in Fig. 7, the method of the present invention can be implemented not only in a graphics API but also at an application level. That is, although the graphics processing is executed through API, the present method can be also implemented for the application level as the step following API. One skilled in the art could apply the present invention to the application level in various ways.

CLAIMS

1. A method of generating on the display screen of a computer a representation of a plurality of three dimensional objects including both opaque and semitransparent objects, information about the apparent depth of points making up an object on the screen being stored in an updatable z-buffer, said method comprising the steps of:

in a first pass, drawing at least one opaque object, updating the Z-buffer and removing hidden surfaces by reference to the contents of the Z-buffer;

in a second pass, drawing a plurality of semi-transparent objects without updating the Z-buffer but removing hidden surfaces by reference to the contents of the Z-buffer from the first pass, the drawing step including blending any of the semitransparent objects which overlie the opaque object; and

in a third pass, updating the Z-buffer, and blending only the semi-transparent objects which overlie other semitransparent objects.

2. A method for generating on the display screen of a computer a representation of a plurality of three dimensional objects including opaque objects and semitransparent objects, information about the apparent depth of points making up an object on the computer display screen being stored in an updatable Z-buffer, said method comprising the steps of:

in a first pass, drawing an opaque object, while updating said Z-buffer and executing hidden surface removal according to an algorithm, making use of said Z-buffer depth information;

in a second pass, drawing a semitransparent object without updating said Z-buffer and while executing hidden surface removal according to said algorithm; and

in a third pass, drawing a semitransparent object, while updating said Z-buffer and executing hidden surface removal according to said Z-buffer algorithm.

3. A method according to claim 2, wherein said steps of drawing said semitransparent object include alpha blending.

4. Computer display apparatus capable of displaying, in response to image data, a representation of a plurality of three-dimensional objects

including opaque objects and semitransparent objects on a display screen, said apparatus comprising:

an updatable Z-buffer for storing apparent depth information of points making up an object on the computer display screen;

an updatable frame buffer for storing the image data to be displayed at each point on the computer display screen; and

a rendering engine, for receiving new image data from a computer and including means for comparing the depth information on the new image data to the depth information that is already stored in said Z-buffer, thereby to determine whether or not the depth of points of the new image data is greater than the depth of the same points already stored in said Z-buffer, means for selectively outputting image data to the frame buffer while updating depth information in the Z-buffer or outputting image data to the frame buffer without updating the depth information, and means for reading the data to be displayed that is already stored in said frame buffer and then blending the read data with the image data received thereafter for each point on the computer display screen.

5. Apparatus according to claim 4, wherein the blending for said semitransparent object is executed by alpha blending.

6. Apparatus according to claim 4 or claim 5 which is a raster scan display.

The Patent Office

INVESTOR IN PEOPLE

16

Application No: GB 9928984.5
Claims searched: 1-6

Examiner: Matthew Males
Date of search: 6 October 2000

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): H4T (TBBA, TBBX, TCHD, TCHX)

Int Cl (Ed.7): G06T 15/00, 15/10, 15/40, 15/50

Other: Online databases: WPI, EPODOC, JAPIO, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
A	EP 0,553,973 A2	(IBM) - whole document but see abstract.	-
A	EP 0,455,374 A2	(IBM) - whole document but see abstract; Figure 2.	-
A	WO 90/02990 A1	(SILICON GRAPHICS INC) - whole document but see abstract.	-
X	US 5,923,333	(HEWLETT PACKARD) - whole document but see column 3, lines 40-43; column 4, line 46-column 6, line 42; Figures, in particular Figures 2-4.	1-6

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.